Icahn
School of
Medicine at
Mount
Sinai

Introduction to coding with R

Part II

Joselyn Chavez

04/19/2022

Let's recap

- Vectors
- One dimension
- All vector components must be the same type
- R finds a way to unify data type
- Define a vector with c()
- Access vector elements using integer index, name or logical evaluation

Data structures in R

- Vectors
- Matrices
- Data frames
- Lists
- Functions

Vectors (Part II)

How to modify a vector?

- Adding a new element

```r
x <- c("a","b","c")
x
```

```
## [1] "a" "b" "c"
```

```r
x[4] <- "d"
x
```

```
## [1] "a" "b" "c" "d"
```

- Removing an element

```
x
```

```
## [1] "a" "b" "c" "d"
```

```
x[-2]
```

```
## [1] "a" "c" "d"
```

```
x <- x[-2]
x
```

```
## [1] "a" "c" "d"
```

Excercise

Using the vector:

```r
fruits <- c("apples" = 1,
            "cherries" = 10,
            "mangos" = 7)
```

- Add a new fruit to the vector
- Remove the cherries

- Replacing an element by index

```
x
```

```
## [1] "a" "c" "d"
```

```
x[1] <- "m"
x
```

```
## [1] "m" "c" "d"
```

- Replacing an element by logical evaluation

```r
x
```

```
## [1] "m" "c" "d"
```

```r
x[x == "d"] <- "e"
x
```

```
## [1] "m" "c" "e"
```

Excercise

Using the vector:

```r
fruits <- c("apples" = 1,
            "cherries" = 10,
            "mangos" = 7)
```

- Select all fruits with values bigger than 5
- Replace the apples number with 4

Matrices

Creating a matrix

Matrices are objects with elements arranged in a two-dimensional layout.

```r
my_matrix <- matrix(data = 1:12, nrow = 4)
my_matrix
```

```
##      [,1] [,2] [,3]
## [1,]    1    5    9
## [2,]    2    6   10
## [3,]    3    7   11
## [4,]    4    8   12
```

- rows and columns
- All elements must be the same type

Operations with matrices

Arithmetic operations

```
my_matrix + 10
```

```
##      [,1] [,2] [,3]
## [1,]   11   15   19
## [2,]   12   16   20
## [3,]   13   17   21
## [4,]   14   18   22
```

```
my_matrix * 2
```

```
##      [,1] [,2] [,3]
## [1,]    2   10   18
## [2,]    4   12   20
## [3,]    6   14   22
## [4,]    8   16   24
```

Operations with matrices

```
matrix1 <- matrix(1:6, nrow = 2, ncol = 3)
matrix2 <- matrix(7:12, nrow = 2, ncol = 3)
```

```
matrix1
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
matrix2
```

```
##      [,1] [,2] [,3]
## [1,]    7    9   11
## [2,]    8   10   12
```

```
matrix1 + matrix2
```

```
##      [,1] [,2] [,3]
## [1,]    8   12   16
## [2,]   10   14   18
```

Data Frames

What is a data frame?

- Two-dimensional arranged data (tables)
- rows and columns
- All columns must be the same length
- Columns can have different type of data
- All components in the column must be the same type (vector)

Creating a data frame

```
fruits <- data.frame(
  name = c("apples","berries","mangos","banar
  number = c(1, 10, 7, 2),
  edible_shell = c(TRUE,TRUE,FALSE,FALSE) )
fruits
```

```
##      name number edible_shell
## 1  apples      1         TRUE
## 2 berries     10         TRUE
## 3  mangos      7        FALSE
## 4 bananas      2        FALSE
```

Properties of data frames

- nrow

```
nrow(fruits)
```

```
## [1] 4
```

- ncol

```
ncol(fruits)
```

```
## [1] 3
```

- dim

```
dim(fruits)
```

```
## [1] 4 3
```

- rownames

```
rownames(fruits)
```

```
## [1] "1" "2" "3" "4"
```

- colnames

```
colnames(fruits)
```

```
## [1] "name"         "number"       "edible_shell"
```

Practice excercise

Create a data frame (patients) that contains the following information:

```
##   first_name last_name age co_morbidity
## 1        Ava     Smith  65         TRUE
## 2       Noah   Johnson  20        FALSE
## 3     Olivia  Williams  47        FALSE
```

- How many rows and columns does the data frame have?
- Print the columns names

How to access data frame elements?

Using row and column index

Syntaxis: df[row,column]

```
fruits
```

```
##      name number edible_shell
## 1  apples      1         TRUE
## 2 berries     10         TRUE
## 3  mangos      7        FALSE
## 4 bananas      2        FALSE
```

```
fruits[2,3]
```

```
## [1] TRUE
```

Select rows 1 to 2 from column 3

```r
fruits[1:2,3]
```

```
## [1] TRUE TRUE
```

Your turn!

Using the patients data frame

```
##   first_name last_name age co_morbidity
## 1        Ava     Smith  65         TRUE
## 2       Noah   Johnson  20        FALSE
## 3     Olivia  Williams  47        FALSE
```

- Extract the last name and age from Ava and Noah

Select all rows from column 2

```
my_matrix[,2]
```

```
## [1] 5 6 7 8
```

Select all columns from row 2

```
my_matrix[2,]
```

```
## [1]  2  6 10
```

Your turn!

Using the patients data frame

```
##   first_name last_name age co_morbidity
## 1        Ava     Smith  65         TRUE
## 2       Noah   Johnson  20        FALSE
## 3     Olivia  Williams  47        FALSE
```

- Extract all the information (columns) from Olivia.
- Extract the age from all patients

Lists

What is a list?

- A collection of like or unlike objects
- Each object can have different dimensions

```
list(a_number = 15,
     fruits = data.frame(name = c("apples",
                                  "mangos"),
                         number = c(3,10)),
     my_matrix = matrix(1:6, nrow = 2)
)
```

```
## $a_number
## [1] 15
##
## $fruits
##     name number
## 1 apples      3
## 2 mangos     10
##
## $my_matrix
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

Thanks!

Ilustration by Allison Horst